*electronics*

MDPI

*Article*

# Promoting Unified Generative Framework with Descriptive Prompts for Joint Multi-Intent Detection and Slot Filling

**Zhiyuan Ma [1,2,3,*,†], Jiwei Qin [1,†], Meiqi Pan[1], Song Tang[1], Jinpeng Mi [1] and Dan Liu [1]**

[1] Institute of Machine Intelligence, University of Shanghai for Science and Technology, Shanghai 200093, China; 213332821@st.usst.edu.cn (J.Q.); 222302289@st.usst.edu.cn (M.P.); tangs@usst.edu.cn (S.T.); jp.mi@usst.edu.cn (J.M.); liudan1123@usst.edu.cn (D.L.)
[2] School of Intelligent Emergency Management, University of Shanghai for Science and Technology, Shanghai 200093, China
[3] State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China
* Correspondence: yuliar3514@usst.edu.cn
† These authors contributed equally to this work.

**Abstract:** Natural language understanding is a crucial aspect of task-oriented dialogue systems, encompassing intent detection (ID) and slot filling (SF). Conventional approaches for ID and SF solve the problems in a separate manners, while recent studies are now leaning toward joint modeling to tackle multi-intent detection and SF. Although the advancements in prompt learning offer a unified framework for ID and SF, current prompt-based methods fail to fully exploit the semantics of intent and slot labels. Additionally, the potential of using prompt learning to model the correlation between ID and SF in multi-intent scenarios remains unexplored. To address the issue, we propose a text-generative framework that unifies ID and SF. The prompt templates are constructed with label semantical descriptions. Moreover, we introduce an auxiliary task to explicitly capture the correlation between ID and SF. The experimental results on two benchmark datasets show that our method achieves an overall accuracy improvement of 0.4–1.5% in a full-data scenario and 1.4–2.7% in a few-shot setting compared with a prior method, establishing it as a new state-of-the-art approach.

**Keywords:** natural language understanding; large language model; intent detection; slot filling



**Citation:** Ma, Z.; Qin, J.; Pan, M.; Tang, S.; Mi, J.; Liu, D. Promoting Unified Generative Framework with Descriptive Prompts for Joint Multi-Intent Detection and Slot Filling. *Electronics* **2024**, *13*, 1087. https://doi.org/10.3390/electronics13061087

Academic Editor: Arkaitz Zubiaga

Received: 31 January 2024
Revised: 9 March 2024
Accepted: 11 March 2024
Published: 15 March 2024



## 1. Introduction

Natural language understanding (NLU) plays a irreplaceable role in task-oriented dialogue (TOD) systems. It helps a machine to understand the user by extracting intent and semantic constituents from users' utterances. A robust NLU not only impacts the performance of downstream tasks, e.g., dialogue state tracking (DST) [1] but also drives advancements in interactive applications such as Siri , Cortana, Alexa, etc. In recent years, there has been a significant increase in research attention paid to NLU [2–11].

A typical approach to NLU involves two separate tasks: intent detection (ID) and slot filling (SF). The former focuses on classifying the intent(s) of an utterance, capturing its sentence-level semantics, while the latter extracts fine-grained information from the utterance in terms of slot–value pairs. With the advancements of NLU studies, there has been a shift toward addressing multi-intent scenarios, where users express multiple intents within a single utterance. This poses additional challenges, particularly in handling the relationship between intent and slot [5,11].

Figure 1 illustrates an example of such an NLU scenario, in which the model is expected to produce intents and their corresponding slot values. Unlike single-intent NLU, each intent in a multi-intent scenario has its own scope. For example, in the phrase "list flights between Pittsburgh and Milwaukee", the intent "atis_flight" pertains to the query within phrase, while "how many Canadian airlines" corresponds to the intent "atis_quantity". Consequently, each slot value with its associated intent is influenced by the semantics

within its respective scope. Therefore, accurately determining each intent and enhancing SF within the respective scope presents a major challenge.

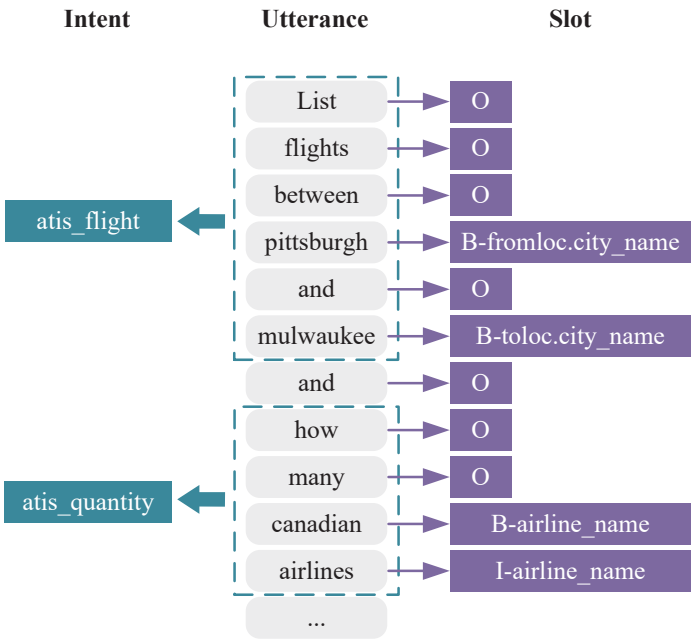



**Figure 1.** An example of multi-intent detection and slot filling. The column (Utterance) refers to the sentence. There are two intents, with their associated sequences in dashed boxes. The column (Slot) refers to the tag sequence. For instance, the tag "B-fromloc.city_name" refers to the first token of slot "fromloc.city_name".

Recent studies [4,12] have confirmed that ID results have a positive impact on SF performance, indicating the two tasks are strongly correlated. In response to these findings, researchers [9,10] have started exploring joint models for ID and SF to leverage shared semantics. Two common approaches involve either using a shared text encoder [6] or employing a masked language model strategy in a pretrained language model (PLM). Additionally, the emergence of large PLMs has facilitated the development of end-to-end solutions in a straightforward manner. One promising paradigm in natural language processing (NLP), called "pretrain-prompt-predict", has been proven successful across various NLP applications [13]. In the context of multi-intent SLU, some researchers [10,14] have adapted masked language techniques in a question-answering task for prompt learning. It maintains the benefits of traditional prompt learning while incorporating prior knowledge (such as the correlation between target and queries) during the fine-tuning process. However, current prompt-based methods lack adequate modeling for the correlation between tasks while constructing the template. Furthermore, the conventional technique of transforming ground truth labels into words lacks domain knowledge and does not fully explore the potential of existing PLMs. Therefore, it is imperative to explore a more unified framework that can effectively capture and utilize domain knowledge to enhance the correlation between ID and SF.

To address the aforementioned issue, we followed the work of UGEN [9] and developed a unified generative framework with descriptive prompt (UGen-DP). It transforms ID and SF into a unified text generation task and enhances the correlation between ID and SF by exploiting the potential semantics. The main contributions of our paper can be summarized as follows:

- We develoepd a prompt construction method with instructional description to enrich the semantics of both intent and slot labels. The approach harnesses the power of PLMs to refine the prompt template, while exploring the benefits of task relation, as showcased in [15].

- To model the correlation between specific intent and slot values, we introduced an auxiliary task called intent-driven slot-filling. It encourages PLMs to capture the inherent correlations between intents and slots, thereby enhancing the overall performance of both ID and SF.
- We conducted extensive experiments on two multi-intent datasets, and we compared our method with current state-of-the-art (SOTA) methods to illustrate the effectiveness and superiority of our method.

## 2. Related Work

### *2.1. Natural Language Understanding*

As previously introduced in Section 1, NLU consists of ID and SF. Early studies investigated the two tasks separately. Classic ID is typically formulated as a text classification problem. Early approaches [16,17] relied on specific model architecture (such as long short-term memory, gated recurrent unit, or capsule networks) to obtain the utterance-level representations. For SF, the problem is framed as a sequence labeling task. Researches have often combined recurrent neural network (RNN)-based models with various attention mechanisms [18–20]. Later, with the development of pretrained language models (PLMs), conventional embeddings (e.g., GloVe) were combined with PLM output to improve the overall recognition performance [21].

Considering that an utterance in dialogue often contains multiple intents, recently, researchers have begun to extend conventional ID to tackle multilabel problems [22]. Kim et al. [23] proposed an approach by breaking the problem into ID in subsentences. However, the dividing of subsentences is challenging when training data are scarce. To mitigate the effects of few annotated data, semisupervised learning [24,25] and metric learning [26] were used. Aiming at the scenario where data are inaccessible, Wang et al. [27] adopted prompt learning with only a pretrained model and few annotated data to achieve ID. To compensate for the domain information between utterances, researches have either focused on adaptively incorporating prior experience and domain-specific knowledge [28] or have constructed a specific network to capture the semantic interactions between utterances [29]. In addition, with the emergence of large language models, prompt learning has also been applied to the task. For instance, Wu et al. [9] and Song et al. [10] used a generative model to learn shared representations across multiple domains and achieved competitive results in both ID and SF.

### *2.2. Joint Model in Natural Language Understanding*

Considering the relationship between ID and SF, joint models have become the mainstream in the field [30]. One straight forward idea, known as the implicit joint mode) [6,7,31,32], involves using a shared encoder to model the feature across two tasks. To name a few, Gangadharaiah et al. [5] proposed a slot gate mechanism for both sentences and tokens; Qin et al. [4] proposed stack propagation to directly use intent information to capture semantics; Qin et al. [6] proposed a graph-based method, where the intent and slot are treated as nodes to model the interaction; Zhu et al. [33] constructed a multigrained graph for dynamic interaction between intents and slots, leading to a better representation. The fusion of intent and slot features improves slot prediction by utilizing the intent information to refine the slot output. In contrast, Xing et al. [32] emphasized the semantic interactions between intent and slot by employing a bidirectional graph. However, they overlooked the semantic meaning of intent labels, which provide valuable slot-related information. To accelerate the speed of inference, Qin et al. [7] introduced a nonautoregressive framework, which enables faster generation and maintains accurate predictions. Zhang et al. [11] jointly modeled ID and SF by incorporating a shared word-level encoder. However, the predicted intent was not utilized to assist with slot generation.

In recent advancements, researchers have shifted their attention to explicitly modeling the interactions between ID and SF. Instead of using shared features, Song et al. [34] used two encoderd (task-shared and task-specific) to encode intent and slot. A graph

neural network was also utilized to directly model the interactions. To adaptively model the interaction, Hou et al. [35] proposed a similarity-based learning framework, and Cai et al. [36] incorporated a slot–intent classifier. Different from our method, these approaches primarily focus on the model structure; researchers have not fully explored the potential of PLMs.

### *2.3. Prompt Learning*

Prompt learning (PL) has gained significant attention with the emergence of large PLMs like GPT-3 and ChatGPT [37]. This paradigm involves transforming original tasks into a text-to-text generation framework, yielding promising results in various NLP directions [1,13,38].

In the field of NLU, PL has garnered increasing interest [9,10,39]. A typical approach involves using masked sentences as prompt templates, where the model predicts the masked tokens based on their associated labels. Jin et al. [39] introduced an instance-aware prompt method that provides a unique template for each sentence. To fully explore the potential of PLM, Wu et al. [9] designed a unified generative framework (UGEN). UGEN transforms conventional masked prediction task into a question-answering paradigm, achieving state-of-the-art (SOTA) performance. With a similar motivation, Song et al. [10] developed an alternative unified framework by converting original utterances into task-specific templates. They utilized intent information to drive slot predictions, implicitly capturing the correlation between intent and slot. However, these methods primarily focus on modeling the relationship between slot type and its value, neglecting the importance of explicitly considering the correlation between specific intents and their corresponding slots. In related topics, such as dialogue state tracking (DST), Gao et al. [1] used prompt templates to emphasize the correlation between keywords and domains. The idea sheds light on constructing specific tasks to model the correlations between intents and slots.

## 3. Unified Generative Framework with Descriptive Prompt

In this section, we first describe the NLU task formulation using a generative framework, which is followed by an overview of our method. Then, the implementation details of label semantic description and intent-driven slot filling are provided.

### *3.1. Task Formulation*

Following the work of UGEN [9], our method transforms both intent detection (ID) and slot filling (SF) into a question-answering paradigm. Specifically, given a user utterance $X = \{w_1, w_2, \ldots, w_n\}$ of length $n$, the ID task involves generating a subset of intent $I_k \in I$, while the SF task involves producing slot–value pairs $S = \{w_i, \cdots w_{i+k} : s_j | w_i, \cdots w_{i+k} \in X, s_j \in S\}$. $I$ and $S$ are the set of possible intents and slot types, respectively.

### *3.2. Framework Overview*

Figure 2 provides an overview of a unified generative framework with descriptive prompt (UGen-DP). Differing from conventional NLU approaches, the question-answering paradigm transforms utterances into questions by adding additional context to enhance the semantic information. For instance, Q1 in Figure 2 is an ID question. The prompt is constructed as "Please identify the intent(s) in the following sentence $\cdots$" to distinguish Q1 from other type of questions. To restrain the answer to the set of $I$, an option that lists all the possible intents is also included in the question (see the yellow rectangle in Figure 2). In this manner, different tasks are unified into the same paradigm, thus allowing a single generative model to jointly learn all tasks.

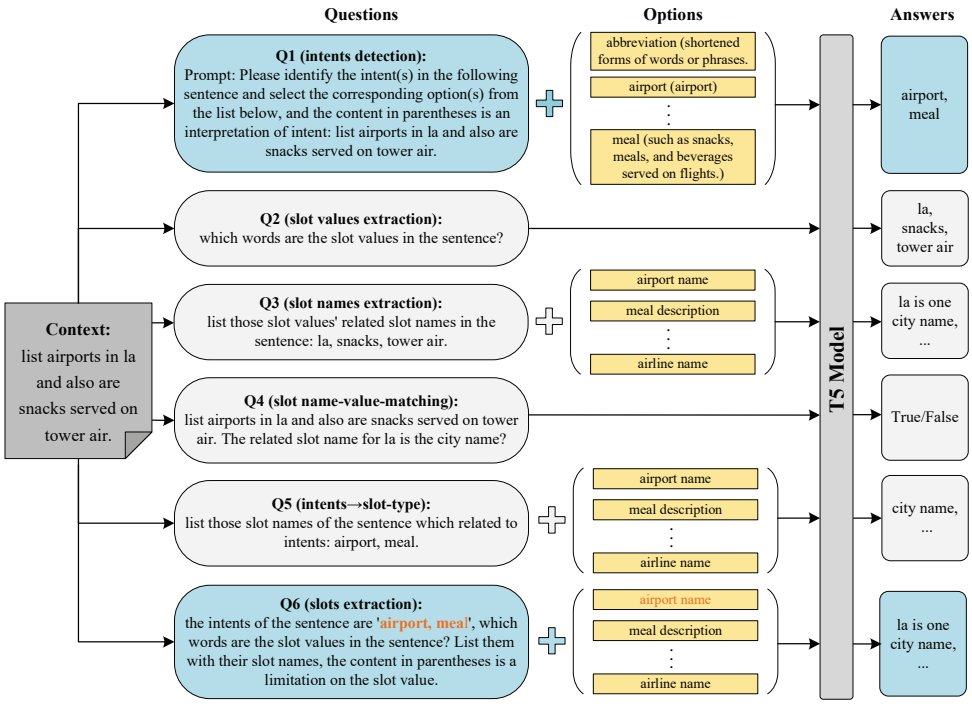



**Figure 2.** Overview of UGen-DP. For a given utterance, UGen-DP updates backbone model with all 6 prompt templates. For inferring phase, only Q1 and Q6 are used to produce the intent and slot.

In UGEN [9], the authors claimed that the quantity of slot types is far larger than that of intents. Therefore, they added 3 additional subtasks to enhance the extraction of slot–value pairs. However, Q1 has no explicit relationship with the other questions, which further hampers the consistency between intent and its corresponding slot–value pairs. Aiming at this issue, we extended UGEN by adding an auxiliary task called intent-driven slot filling, which emphasizes the correlation between intents and slots (see Q5). Accordingly, Q6 is extended by incorporating potential intent label(s) into the prompt template (see the red phrase in Q6). The intents in Q6 are the intent label(s) of the utterance; we used ground truth labels for training and the output of Q1 for inference.

Another difference between UGen-DP and UGEN is that we enhance the performance of ID and SF by incorporating label semantical descriptions for both intent and slot labels. In reproducing the results of UGEN, we found that some of the labels (either intent in Q1 or slot types in Q2) were misclassified, especially for those that were rare in training set. We believe it is insufficient to solely rely on la arge model to comprehend domain-specific labels. Therefore, we added a descriptivel prompt for each label to enhance their semantics.

Before training, UGen-DP first transforms the original utterance into question–answer pairs with all 6 templates, as shown in Figure 2. To train the model jointly on these tasks, we integrate all the losses from Q1 to Q6 as $L$. For each task, UGen-DP minimizes the negative log-likelihood loss (see Equation (1)), which is similar to other text-to-text methods. $\theta$ is the learnable model parameter, $y_i$ is $i$th predicted token, and $|Y|$ refers to the sequence length.

$$\begin{aligned} L &= \sum_{j=1}^{6} L_j \\ L_j &= -\sum_{i=1}^{|Y|} \log p(y_i|X, \theta) \end{aligned} \tag{1}$$

### 3.3. Label Semantic Description

As mentioned in Section 3.2, a large language model is employed to produce multi-intent labels [9,10]. It has been shown that expanding the semantical meaning of label can

lead to improved overall performance [10,14]. However, current methods mainly focus on expanding abbreviations or converting tokens into words, which lacks specific semantic construction for domain intent labels. Therefore, we aimed to construct a semantic description for each label, allowing the large model to capture more label-related information.

Table 1 presents a sample list of intents and their corresponding descriptions. Instead of explicitly removing a special token (such as "_") from the labels, we additionally added a description phrase at the end of each label. Specifically, we first extracted the words associated with different intent labels from the training set and selected the most frequent ones as the initial description.

**Table 1.** Examples of intents with their corresponding semantic descriptions.

| Original Intent | Intent with Semantic Description |
|---|---|
| atis_abbreviation | abbreviation (shortened forms of words or phrases) |
| atis_airport | airport (airport) |
| atis_city | city (from somewhere to somewhere, such as cities, locations) |
| atis_capacity | capacity (such as seats) |

For labels with specific meanings, such as "atis_airport", we used the original label as the description. For labels containing one or more concepts, we enriched the semantics by including their subclasses. For example, the description for "meal" can be expanded as "meal (such as snacks, meals, and beverages served on flights)". After preprocessing the data, the prompt template for Q1 can be formatted as "<s> Prompt: [Template]:[Utterance] </s> Options: [Intent List] </s>". Here, "Template" refers to the prompt sentence between the utterance and the word "Prompt:". "Intent List" includes the intent labels with semantic descriptions.

With the success of ChatGPT and other similar pretrained generative models, large models can memorize the context of a conversation and generate topic-related answers. These characteristics make it appealing to explore the possibility of incorporating the external knowledge (specific label semantics) contained in such large models. Motivated by this idea, we refined the prompt template via a multistep conversation with large model.

Table 2 provides a simple demonstration of refining the template. In the training set, there is an utterance "From Boston to BWI that serves a snack...", where the word "snack" is tagged as "B-meal_description". We utilized the word and the intent label "atis_meal" (colored in blue) to stimulate the text-generative model (ChatGPT) to produce words and phrases related to "snack" and "meal". Subsequently, we further guided the model to generate text containing the prompt description.

In the end, the final prompt template was as follows: "<s>Please identify the intent(s) in the following sentence and select the corresponding option(s) from the list below. The content in parentheses provides an interpretation of the intent: </s> [Utterance]</s> Options: [Intent List]</s>".

To enhance the semantic description of slot labels, we performed a similar but simpler operation to generate additional semantic descriptions. Table 3 presents a sample list of slots, which were randomly selected from the dataset. We first constructed a baseline using UGEN and summarized its performance on the training set. For slots with high accuracy, we simply converted the labels into phrases by removing special tokens such as "_".

**Table 2.** An example of using a large language model to refine semantic descriptions. "Q" refers to the question for large language model, which is the input from the user, while "R" is the response of the large model.

| | |
|---|---|
| Q: | In common understanding, when the word "snacks" appears in a sentence, there is a high probability that the intention is to refer to "atis_meal". Consequently, the semantic information of labels can be enhanced through semantic expansion. |
| R: | · · · For instance, you can gather words, phrases, or concepts related to meals, such as "airline meals", "pilot meals", "special diets", "appetizers", "beverages", "snacks", and so on, and add them to the semantic information of the "atis_meal " intent label. · · · |
| Q: | According to the description above, the intent label is "atis_meal". Please return an optimized semantic expansion of the label (separated by a colon). |
| R: | Sure, here is an example of expanded "atis_meal" label based on the semantic information: "atis_meal": snacks, meals, and beverages served on flights. |

For other slots with prepositions, we added a prepositional combination as the description, e.g., from location.city name (slot value before the word 'to'). As for the slots with vague meanings, we added descriptions by either using a large language model (similar to the procedure for intent) or excluding the negative examples, depending on whether the response of large language model is meaningful. For instance, the slot "flight_mod" in Table 3 cannot be enumerated using the response from a large language model. In this case, we summarized the negative samples by using the original UGEN and used the phrase "day of the week" to represent the majority of negative samples.

**Table 3.** Examples of slot label with their corresponding semantic descriptions. "Original Slot" refers to the slot name in the dataset. Constructed slot names for UGEN and our method are provided in "Baseline" and "Slot with Semantical Descriptions", respectively.

| **Original Slot** | **Baseline** | **Slot with Semantic Descriptions** |
|---|---|---|
| flight_number | flight number | flight number |
| fromloc.city_name | from location.city name | from location.city name (slot value before the word 'to') |
| flight_mod | flight mod | flight mod (slot value excluding 'day of the week') |
| depart_date.today _relative | depart date.today relative | depart date.today relative |

### *3.4. Intent-Driven Slot Filling*

In UGEN [9], SF is divided into four subtasks: (1) slot value extraction, (2) slot name extraction, (3) slot name–value matching, and (4) slot extraction. The first two subtasks aim to help the PLM learn the correlation between tokens and slot names (values). The third subtask is designed to build the relationship between slot names and values, while the last subtask ensures that the model can correctly provide slot names and their corresponding values. Although UGEN unifies both ID and SF in the same framework, the subtasks are separately processed. Slot types that are not in the scope of the true intent are inevitably generated.

In the work of Song et al. [10], an auxiliary subtask was used to encourage semantical interactions among tokens, intents, and slots, thereby strengthening the correlation between different tasks. However, this modeling approach ignores the direct correlation between specific intents and their corresponding slots. A more straightforward idea involves constructing a subtask to explicitly using the result of ID for SF. Motivated by this idea, we propose another subtask called intent-driven slot filling (see Q5 and Q6 in Figure 2). It adds another auxiliary task to predict the associated slot types for a given intent. Subsequently,

it explicitly incorporates the predicted intent from Q1 and provides the corresponding slot types. In this manner, it forces the PLM to focus on the correlation between each intent and its associated slot types.

The overall prompt template is as follows: "<s> Sentence: The intents of the sentence are *I*. Which words are the slot values in the sentence? List them with their slot names. The content in parentheses provides a limitation on the slot values:</s> Options: *S* </s>". Here, *I* refers to the predicted intent(s), and *S* represents the set of corresponding slot types related to *I*. In the training process, we used the ground truth labels to construct the question, while, in the inference stage, we used the output of Q1 to construct the template.

## 4. Experiments

In this section, we first introduce the dataset and experimental settings. Then, we evaluate our method against recent multi-intent and slot filling approaches.

### *4.1. Datasets and Settings*

To evaluate the performance of UGen-DP, we conducted comprehensive experiments on two challenging and widely used datasets: *MixSNIPS* and *MixATIS*. MixSNIPS was constructed based on SNIPS [40], while MixATIS was built on the ATIS [41]. The statistics of the datasets are shown in Table 4. As a challenging NLU dataset, MixATIS has only 17 out of 18 intents in the training utterance (missing intent "day name"). The test set contains 16 intents (missing "cheapest" and "restriction"). A similar issue exists for slot labels. The difference between the train and test set explains why most methods achieve better performance on MixSNIPS than on MixATIS.

**Table 4.** Statistics of MixATIS and MixSNIPS. "#" refers to the number of each value.

| **MixATIS** | | **MixSNIPS** | |
|---|---|---|---|
| Train (#) | 13,161 | Train (#) | 39,776 |
| Val (#) | 759 | Val (#) | 2198 |
| Test (#) | 828 | Test (#) | 2199 |
| Intent (#) | 18 | Intent (#) | 7 |
| Slot (#) | 78 | Slot (#) | 39 |

All experiments were conducted on a Linux X64 Server with a 64 GB NVIDIA A6000 GPU. For comparison purposes, we used the T5 base model as the backbone (generative model), which consisted of 12 encoder/decoder layers. Other implementation details are summarized in Table 5. It is worth noting that UGen-DP achieved stable performance before 30 epochs, so we used this setting for all experiments.

**Table 5.** Experimental settings.

| **Experimental Settings** | | | |
|---|---|---|---|
| Backbone | T5-base | Hidden Size | 768 |
| Optimizer | Adam | Learning Rate | $1 \times 10^{-5}$ |
| Batch Size | 4 | No. of Epochs | 30 |

### *4.2. Overall Comparison Results*

To demonstrate the improvements of UGen-DP, we compare it with 10 current approaches in NLU. Table 6 summarizes the results on MixATIS and MixSNIPS. For ID and SF, we use accuracy ("I-Acc") and F1 score ("S-F1"), respectively. Moreover, we also measure the overall performance in terms of accuracy ("O-Acc").

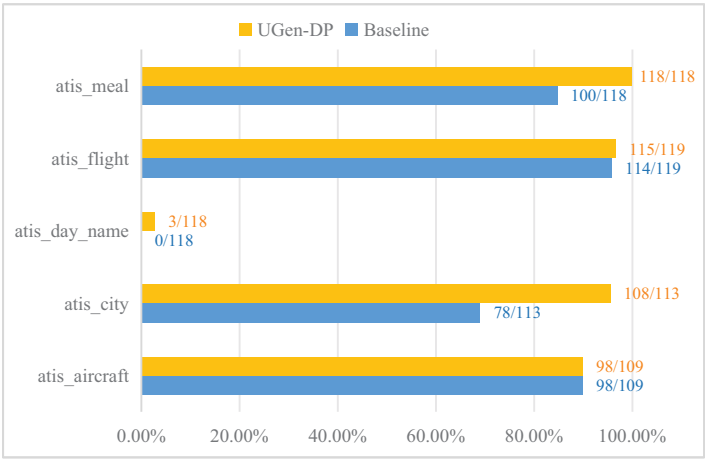
**Table 6.** Overall results on the *MixSNIPS* and *MixATIS* datasets with full data. * refers to the best result presented in [10]. "S-F1" is the F1 scores of slot filling, while "I-Acc" and "O-Acc" refer to the accuracy of intent prediction and overall performance (both intents and slots are correct), respectively. The best scores are highlighted in bold.

| Model | MixATIS | | | MixSNIPS | | |
|---|---|---|---|---|---|---|
| | S-F1 | I-Acc | O-Acc | S-F1 | I-Acc | O-Acc |
| Bi-Model [2] | 83.9 | 70.3 | 34.4 | 90.7 | 95.6 | 63.4 |
| SF_ID [3] | 87.4 | 66.2 | 34.9 | 90.6 | 95.0 | 59.9 |
| Stack-Propagation [4] | 87.8 | 72.1 | 40.1 | 94.2 | 96.0 | 72.9 |
| Joint Learning [5] | 84.6 | 73.4 | 36.1 | 90.6 | 95.1 | 62.9 |
| AGIF [6] | 86.7 | 74.4 | 40.8 | 94.2 | 95.1 | 74.2 |
| GL-GIN [7] | 88.3 | 76.3 | 43.5 | 94.9 | 95.6 | 75.4 |
| SDJN [8] | 88.2 | 77.1 | 44.6 | 94.4 | 96.5 | 75.5 |
| DGIF [33] | 88.5 | 83.3 | 50.7 | 95.9 | 97.8 | 84.3 |
| PromptSLU [10] | 89.6 | 85.8 | 57.2 | 96.5 | 97.5 | **84.8** * |
| UGEN [9] | 89.2 | 83.0 | 55.3 | 95.0 | 96.9 | 78.8 |
| UGen-DP | **90.3** | **86.2** | **58.7** | **96.6** | **97.6** | **84.7** |

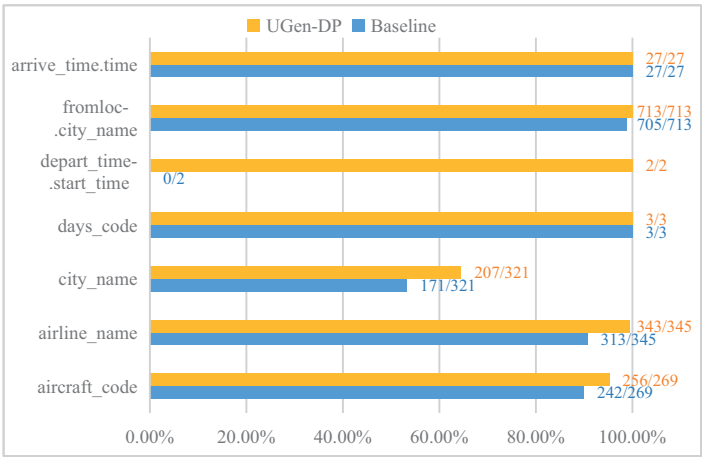
In general, UGen-DP outperformed the other methods except for PromptSLU [10] in O-Acc on MixSNIPS. Compared with UGen, UGen-DP maintained an improvement of 3.2 in I-Acc on MixATIS. Similar results were observed on MixSNIPS. This finding proves that, with semantic descriptions, UGen-DP has better intent identification ability than UGEN. Moreover, UGen-DP surpassed other methods by over 1.5 in O-Acc on MixATIS. For MixSNIPS, although there was a 0.1% gap between PromptSLU and UGen-DP, our method maintained the best results for both datasets. Note that there was no available code for [10]. Hence, we directly used the results reported in [10].

To better illustrate the improvements provided by UGen-DP over UGEN, we evaluated the fine-grained performance on the intent and slot label (see Figure 3). For intent, we drew all 5 intents in the test set (see Figure 3a). For slos, we selected 5 slot categories that were not fully recognized and 2 that were correctly labeled. The other 10 intent labels were predicted by both UGEN and UGen-DP with 100% accuracy; hence, the results for these labels are not included in Figure 3.

In general, our method outperformed UGEN in detecting all intents. For "atis_city" and "atis_meal", UGen-DP detected 108/133 (81.20%) and 118/118 (100%), while UGEN only detected 78/133 (58.64%) and 100/118 (84.74%). For the particular intent "atis_day_name", which did not appear in the training set, UGEN recognized none. However, with the help of the semantic description for the intent label, UGen-DP detected some of the intents. In our experiments, we also observed that although UGen-DP could detect some of the utterances with "atis_day_name", the performance was not stable. Under the same settings, the best detection rate was over 10/118. The reason for this finding may be that there was no training sample to provide correct supervised signals. Therefore, the detection only relied on the semantical description, which could have led to T5 not fully capturing the semantics.




(**a**) Fine-grained results for intent detection.

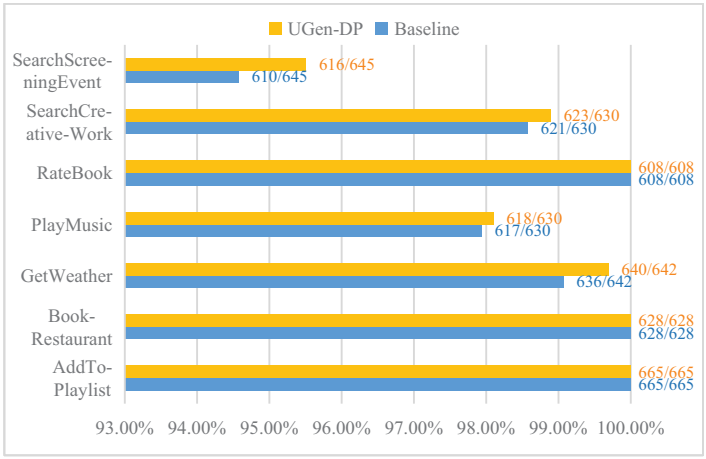



(**b**) Fine-grained results for slot filling.

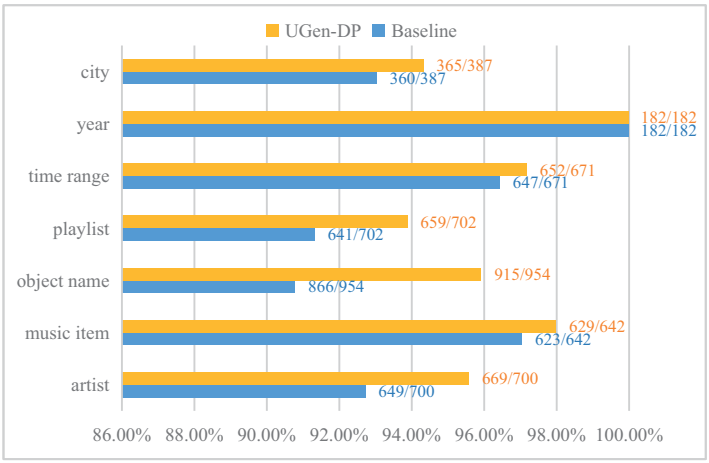
**Figure 3.** Fine-grained results for intent and slot on *MixATIS*. "Baseline" refers to UGEN. (**a**) All 5 intents in the test set; (**b**) 7 slots, including 5 that were not fully recognized and 2 that were correctly predicted.

We also conducted the same evaluation on MixSNIPS, and similar results were observed (see Figure 4). For ID (see Figure 4a), our method maintained equal or better performance than UGEN, but the gap was not as significant as on MixATIS. The reason may be that there are more training samples in the MixSNIPS dataset, and all seven intents existed in both the train and test sets. Therefore, having enough samples leads to better performance. The same results also applied to slot filling (see Figure 4b), except that there were more slot categories. Therefore, the differences between the methods were amplified. For slot categories like "object name", our method obtained an improvement of over 5%.

To track the performance changes in the training process, we summarize the loss and overall accuracy on MixATIS in Figure 5. It shows that with the decrease in training loss, the overall accuracy on the test set increases and eventually converges to a stable state. The results indicate that the decrease in $L = \sum_{j=1}^{6} L_j$ ensures the probability of $P(y_i|X, \theta)$ improving. Moreover, the overall accuracy is relatively stable with no clear performance drop between adjacent epochs.




(**a**) Fine-grained results for intent detection.




(**b**) Fine-grained results for slot filling.

**Figure 4.** Fine-grained results for intent and slot in *MixSNIPS*. "Baseline" refores to UGEN. (**a**) All 7 intent predictions in the test set; (**b**) 6 slot categories that are not fully recognized, and 1 that is correctly predicted.

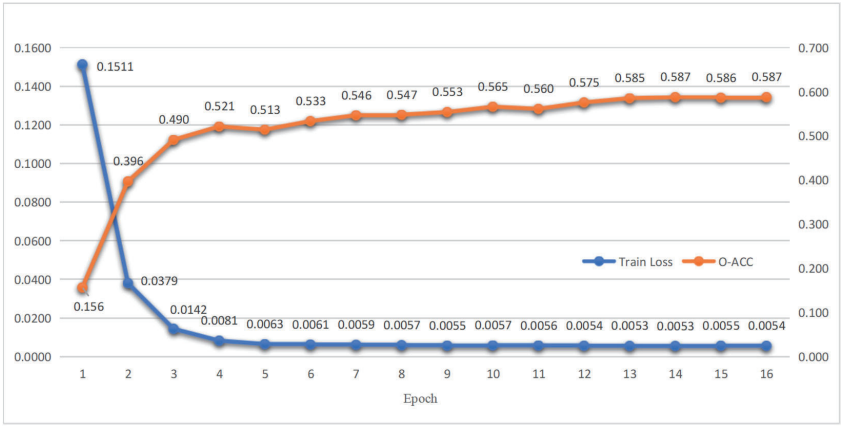



**Figure 5.** Loss and overall accuracy in the training process on MixATIS. The overall accuracy on test set and training loss are colored in orange and blue, respectively.

### *4.3. Comparison in Few-Shot Scenario*

To better demonstrate the effectiveness in a few-shot scenario, we conducted experiments using 5, 10, and 10% of training data to tune the generative model. For fair comparison, we used the same sampling method as in [9] for the experiments. Table 7 summarizes the results.

**Table 7.** Results on the MixSNIPS dataset in few-shot settings. * SP refers to stack propagation. For SP, AGIF, and GL-GIN, we used the results reported in [9]. The best scores are highlighted in bold.

| Model | 5-Shot | | | 10-Shot | | | 10% | | |
|---|---|---|---|---|---|---|---|---|---|
| | S-F1 | I-Acc | O-Acc | S-F1 | I-Acc | O-Acc | S-F1 | I-Acc | O-Acc |
| * SP [4] | 58.7 | 78.2 | 11.9 | 71.5 | 88.3 | 24.8 | 90.3 | 93.5 | 58.4 |
| AGIF [6] | 60.7 | 77.8 | 14.4 | 73.0 | 86.3 | 27.5 | 91.2 | 93.0 | 62.8 |
| GL-GIN [7] | 54.3 | 86.1 | 10.1 | 69.5 | 90.2 | 23.9 | 92.1 | 95.3 | 66.6 |
| UGEN [9] | 84.2 | 92.4 | 42.5 | 87.4 | 93.3 | 50.5 | 93.6 | 96.0 | 71.7 |
| UGen-DP | **85.9** | **93.2** | **43.9** | **89.3** | **94.1** | **52.2** | **94.1** | **96.2** | **74.4** |

With 10% annotated samples, UGen-DP outperformed UGEN in O-Acc by 2.7, although both I-Acc and S-F1 increased by 0.2 and 0.5, respectively. This indicates that UGen-DP can better predict the intent and its related slot when annotated data are scarce. Similar improvements were observed in the 5/10-shot results (increased by 1.4/1.7). However, the difference decreased as the number of training sample dropped. In the meantime, compared with the other methods in five-shot settings, UGen-DP maintained a superiority of over 29.5 (compared with AGIF). Even when 10% of the data was used, the gap was still over 7.8 (compared with GL-GIN).

### *4.4. Ablation Study*

To further investigate the improvements of UGen-DP compared with UGEN, we conducted an ablation study, and we summarize the results in Table 8.

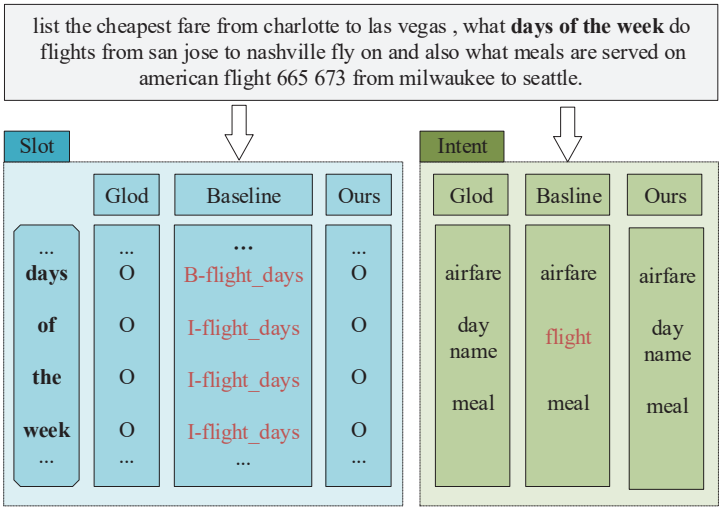
**Table 8.** Ablation study using *MixATIS* and *MixSNIPS*. "LSD" refers to label semantic description, while "IDSF" denotes intent-driven slot filling. If there was no refined template, LSD or IDSF is used; UGen-DP was identical to UGEN. Therefore, we used UGEN as the baseline in the ablation study.

| Model | MixATIS | | | MixSNIPS | | |
|---|---|---|---|---|---|---|
| | S-F1 | I-Acc | O-Acc | S-F1 | I-Acc | O-Acc |
| UGEN | 89.2 | 83.0 | 55.3 | 95.0 | 96.9 | 78.8 |
| w/o LSD or IDSF | 89.5 | 84.1 | 55.9 | 95.7 | 97.1 | 81.2 |
| w/o IDSF | 89.7 | 86.0 | 56.2 | 95.8 | 97.4 | 81.6 |
| w/o LSD | 89.9 | 84.3 | 56.4 | 96.4 | 97.2 | 84.2 |
| UGen-DP | 90.3 | 86.2 | 58.7 | 96.6 | 97.6 | 84.7 |

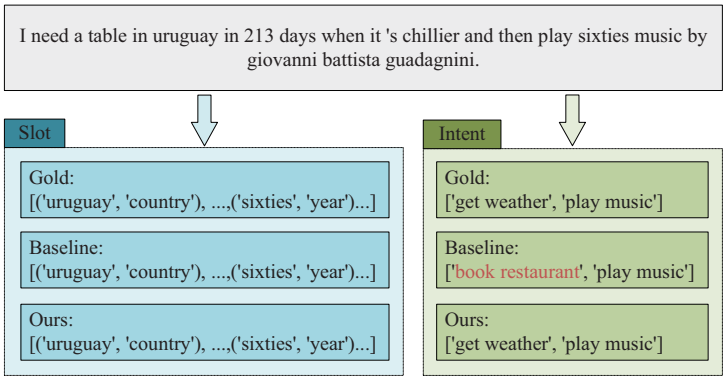
As shown in Table 8, with only the refined prompt template ("w/o LSD and IDSF"), there were slight O-Acc improvements on MixATIS (0.6). However, for MixSNIPS, the improvement reached 2.4. The difference shows that although the large model could provide latent semantics, the effects varied from one domain to another. In the meantime, when label semantic description (denoted by "LSD") was included, the performance increased on both datasets. For MixATIS, it improved 0.2, 1.9, and 0.3 for S-F1, I-Acc, and O-Acc, respectively. With further integration of intent-driven slot filling (denoted by 'IDSF'), the improvement escalated. Without IDSF, the performance on MixATIS ("O-Acc") dropped 2.5 and that on MixSNIPS dropped 3.5. This indicates that combining the intent in slot filling helps the method to learn strong associations between intent and its corresponding slot(s). Hence, the overall performance improves.

### *4.5. Case Study*

To further illustrate the effectiveness of UGen-DP, Figure 6 shows two representative samples and their model predictions. In the first case (a), the sentence contains three intents, i.e., "airfare", "day name", and "meal". UGen-DP correctly predicted all three intents, while UGEN misclassified them into "flight". The wrong intent further caused the phrase "days of the week" to be tagged as "flight days". With label semantical description, the prompt template can provide additional information to help recognize both intents and slots.




(**a**) Case 1: Wrong intent and wrong slot value.




(**b**) Case 2: Wrong intent and correct slot value.

**Figure 6.** Two representative utterances in *MixATIS*. "Baseline" refers to UGEN, and "Gold" is the ground truth. All intent and slot labels were transformed to phrases for comprehensive purposes.

In the second case (b), it can be observed that both our method and UGEN correctly recognized all slots. However, the intent of "get weather" was mis-recognized into "book restaurant" by UGEN. We believe the reason for this is that UGEN only helped to capture the correlation between "need a table" and "book restaurant", while the correlation between the correct intent "get weather" and "it's chillier" was not obtained. We further examined the dataset and found few utterances that contained adjectives related to weather. Therefore, by incorporating a semantical description with the intent "get weather", it helped to capture the correlation. Moreover, since we could not enumerate all adjectives related to weather, we chose a more general way of incorporating similar words that commonly appear in other contexts. In this example, the semantic description contains words such as "warm" and "warmer". By doing so, UGen-DP successfully captured the correlation, as presented in (b).

## 5. Conclusions

In this paper, we proposed a prompt learning framework (called UGen-DP) for joint multi-intent and slot filling, which solves ID and SF in a unified question-answering paradigm. To achieve better performance, we incorporated a semantic description to enhance the semantics of the intent and slot labels. In addition, we constructed a subtask of using intent prediction to promote slot filling. Moreover, we exploited the potential of utilizing a text-generative model to help rephrase the template. Comprehensive experiments on two challenging datasets showed that UGen-DP outperforms other methods and achieves competitive performance in few-shot scenarios. Future directions include automatically

constructing the prompt template based on interactions with a large language model and exploring additional techniques for optimizing performance. For scenarios with no training instances regarding specific intent labels, integrating the zero-shot learning mechanism is also important to improve the generalization ability of the method.

**Author Contributions:** Z.M. was involved in the conceptualization, methodology, formal analysis, original draft, review and editing, and supervision. J.Q. contributed to the methodology, implementation, data curation, and original draft. M.P. contributed to the implementation, data curation, and editing. S.T. contributed to the methodology, validation, and original draft. J.M. assisted with the methodology, formal analysis, and validation. D.L. contributed to the validation, original draft, and review and editing. All authors have read and agreed to the published version of the manuscript.

**Funding:** The research leading to these results received funding from State Key Laboratory for Novel Software Technology, Nanjing University under grant agreement No. KFKT2021B39.

**Data Availability Statement:** Data are contained within the article.

**Acknowledgments:** The authors would like to thank the anonymous reviewers for their insightful comments.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Gao, Q.; Dong, G.; Mou, Y.; Wang, L.; Zeng, C.; Guo, D.; Sun, M.; Xu, W. Exploiting domain-slot related keywords description for few-shot cross-domain dialogue state tracking. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates, 2 December 2022; pp. 2460–2465. [CrossRef]
2. Wang, Y.; Shen, Y.; Jin, H. A Bi-model based RNN semantic frame parsing model for intent detection and slot filling. In Proceedings of the The North American Chapter of the Association for Computational Linguistics, New Orleans, LA, USA, 1–6 June 2018; pp. 309–314. [CrossRef]
3. E, H.; Niu, P.; Chen, Z.; Song, M. A novel bi-directional interrelated model for joint intent detection and slot filling. In Proceedings of the the Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July 2019; pp. 5467–5471. [CrossRef]
4. Qin, L.; Che, W.; Li, Y.; Wen, H.; Liu, T. A stack-propagation framework with token-level intent detection for spoken language understanding. *arXiv* **2019**, arXiv:1909.02188. [CrossRef].
5. Gangadharaiah, R.; Narayanaswamy, B. Joint multiple intent detection and slot labeling for goal-oriented dialog. In Proceedings of the the Conference of the North American Chapter of the Association for Computational Linguistics, Minneapolis, MN, USA, 2–7 June 2019; pp. 564–569. [CrossRef]
6. Qin, L.; Xu, X.; Che, W.; Liu, T. AGIF: An adaptive graph-interactive framework for joint multiple intent detection and slot filling. *arXiv* **2020**, arXiv:2004.10087. [CrossRef].
7. Qin, L.; Wei, F.; Xie, T.; Xu, X.; Che, W.; Liu, T. GL-GIN: Fast and accurate non-autoregressive model for joint multiple intent detection and slot filling. In Proceedings of the the Annual Meeting of the Association for Computational Linguistics and International Joint Conference on Natural Language Processing, Online, 1–6 August 2021; pp. 178–188. [CrossRef]
8. Chen, L.; Zhou, P.; Zou, Y. Joint multiple intent detection and slot filling via self-distillation. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Singapore, 7–13 May 2022; pp. 7612–7616. [CrossRef]
9. Wu, Y.; Wang, H.Q.; Zhang, D.; Chen, G.; Zhang, H. Incorporating instructional prompts into a unified generative framework for joint multiple intent detection and slot filling. In Proceedings of the International Conference on Computational Linguistics, Gyeongju, Republic of Korea, 12–17 October 2022; pp. 7203–7208.
10. Song, F.; Huang, L.; Wang, H. A unified framework for multi-intent spoken language understanding with prompting. *arXiv* **2022**, arXiv:2210.03337. [CrossRef].
11. Zhang, Q.; Wang, S.; Li, J. A heterogeneous interaction graph network for multi-intent spoken language understanding. *Neural Process. Lett.* **2023**, *55*, 9483–9501. [CrossRef]
12. Cheng, L.; Yang, W.; Jia, W. A scope sensitive and result attentive model for multi-intent spoken language understanding. *arXiv* **2022**, arXiv:2211.12220. [CrossRef].
13. Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; Neubig, G. Pre-Train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.* **2023**, *55*, 1–35. [CrossRef]
14. Wang, L.; Li, R.; Yan, Y.; Yan, Y.; Wang, S.; Wu, W.Y.; Xu, W. InstructionNER: A multi-task instruction-based generative framework for few-shot NER. *arXiv* **2022**, arXiv:2203.03903. [CrossRef].
15. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Chi, E.; Le, Q.; Zhou, D. Chain of thought prompting elicits reasoning in large language models. *arXiv* **2022**, arXiv:2201.11903. [CrossRef].

16. Firdaus, M.; Bhatnagar, S.; Ekbal, A.; Bhattacharyya, P. Intent detection for spoken language understanding using a deep ensemble model. In *Proceedings of the PRICAI 2018: Trends in Artificial Intelligence*; Geng, X., Kang, B.H., Eds.; Springer: Cham, Switzerland, 2018; pp. 629–642.
17. Xia, C.; Zhang, C.; Yan, X.; Chang, Y.; Yu, P. Zero-shot user intent detection via capsule neural networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*; Riloff, E., Chiang, D., Hockenmaier, J., Tsujii, J., Eds.; Association for Computational Linguistics: Brussels, Belgium, 2018; pp. 3090–3099. [CrossRef]
18. Shin, Y.; Yoo, K.M.; Lee, S.-g. Slot filling with delexicalized sentence generation. In Proceedings of the Interspeech, Hyderabad, India, 2–6 September 2018; pp. 2082–2086. [CrossRef]
19. Wu, J.; Banchs, R.E.; D'Haro, L.F.; Krishnaswamy, P.; Chen, N. Attention-based semantic priming for slot-filling. In *Proceedings of the the Seventh Named Entities Workshop*; Chen, N., Banchs, R.E., Duan, X., Zhang, M., Li, H., Eds.; Association for Computational Linguistics: Melbourne, Australia, 2018; pp. 22–26. [CrossRef]
20. Zhu, S.; Yu, K. Encoder-decoder with focus-mechanism for sequence labelling based spoken language understanding. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, New Orleans, LA, USA, 5–9 March 2017; pp. 5675–5679. [CrossRef]
21. Qiu, L.; Ding, Y.; He, L. Recurrent neural networks with pre-trained language model embedding for slot filling task. *CoRR* **2018**, arXiv:1812.05199. [CrossRef].
22. Ding, Z.; Yang, Z.; Lin, H.; Wang, J. Focus on interaction: A novel dynamic graph model for joint multiple intent detection and slot filling. In Proceedings of the the International Joint Conference on Artificial Intelligence. International Joint Conferences on Artificial Intelligence Organization, Montreal, QC, Canada, 26 August 2021; pp. 3801–3807. [CrossRef]
23. Kim, B.; Ryu, S.; Lee, G.G. Two-stage multi-intent detection for spoken language understanding. *Multimed. Tools Appl.* **2017**, *76*, 11377–11390. [CrossRef]
24. Kumar, A.; Tripathi, R.K.; Vepa, J. Low resource pipeline for spoken language understanding via weak supervision. *arXiv* **2022**, arXiv:2206.10559. [CrossRef].
25. Yang, F.; Zhou, X.; Wang, Y.; Atawulla, A.; Bi, R. Diversity features enhanced prototypical network for few-shot intent detection. In Proceedings of the International Joint Conference on Artificial Intelligence, Vienna, Austria, 23–29 July 2022; Volume 7, pp. 4447–4453. [CrossRef]
26. Hou, Y.; Chen, C.; Luo, X.; Li, B.; Che, W. Inverse is better! Fast and accurate prompt for few-shot slot tagging. In Proceedings of the Findings of the Association for Computational Linguistics, Dublin, Ireland, 22–27 May 2022; Muresan, S., Nakov, P., Villavicencio, A., Eds.; Association for Computational Linguistics: Melbourne, Australia, 2022; pp. 637–647. [CrossRef].
27. Wang, Y.; Mei, J.; Zou, B.; Fan, R.; He, T.; Aw, A.T. Making pre-trained language models better learn few-shot spoken language understanding in more practical scenarios. In Proceedings of the Findings of the Association for Computational Linguistics, Toronto, ON, Canada, 9–14 July 2023; Rogers, A., Boyd-Graber, J., Okazaki, N., Eds.; Association for Computational Linguistics: Melbourne, Australia, 2023; pp. 13508–13523. [CrossRef]
28. Hou, Y.; Lai, Y.; Wu, Y.; Che, W.; Liu, T. Few-shot learning for multi-label intent detection. In Proceedings of the the AAAI Conference on Artificial Intelligence, Atlanta, GA, USA, 17 May 2021; Volume 35, pp. 13036–13044. [CrossRef]
29. Zhang, F.; Chen, W.; Ding, F.; Wang, T. Dual class knowledge propagation network for multi-label few-shot intent detection. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*; Rogers, A., Boyd-Graber, J., Okazaki, N., Eds.; Association for Computational Linguistics: Melbourne, Australia, 2023; pp. 8605–8618. [CrossRef]
30. Qin, L.; Xie, T.; Che, W.; Liu, T. A survey on spoken language understanding: Recent advances and new frontiers. *arXiv* **2021**, arXiv:2103.03095. [CrossRef].
31. Zhang, X.; Wang, H. A joint model of intent determination and slot filling for spoken language understanding. In Proceedings of the International Joint Conference on Artificial Intelligence, New York, NY, USA, 9–15 July 2016; pp. 2993–2999.
32. Xing, B.; Tsang, I. Co-guiding net: Achieving mutual guidances between multiple intent detection and slot filling via heterogeneous semantics-label graphs. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates, 11 December 2022; pp. 159–169. [CrossRef]
33. Zhu, Z.; Xu, W.; Cheng, X.; Song, T.; Zou, Y. A dynamic graph interactive framework with label-semantic injection for spoken language understanding. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Rhodes Island, Greece, 4–10 June 2023; pp. 1–5. [CrossRef]
34. Song, M.; Yu, B.; Quangang, L.; Yubin, W.; Liu, T.; Xu, H. Enhancing joint multiple intent detection and slot filling with global intent-slot co-occurrence. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates, 11 December 2022; pp. 7967–7977. [CrossRef]
35. Hou, Y.; Lai, Y.; Chen, C.; Che, W.; Liu, T. Learning to bridge metric spaces: Few-shot joint learning of intent detection and slot filling. In Proceedings of the Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Online, 1–6 August 2021; pp. 3190–3200. [CrossRef]
36. Cai, F.; Zhou, W.; Mi, F.; Faltings, B. Slim: Explicit slot-intent mapping with BERT for joint multi-intent detection and slot filling. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Singapore, 7–13 May 2022; pp. 7607–7611. [CrossRef]

37. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. In Proceedings of the Advances in Neural Information Processing Systems, San Francisco, CA, USA, 6–12 May 2020; Volume 33, pp. 1877–1901.
38. Gao, T.; Fisch, A.; Chen, D. Making pre-trained language models better few-shot learners. *arXiv* **2020**, arXiv:2012.15723. [CrossRef].
39. Jin, F.; Lu, J.; Zhang, J.; Zong, C. Instance-aware prompt learning for language understanding and generation. *arXiv* **2022**, arXiv:2201.07126. [CrossRef].
40. Coucke, A.; Saade, A.; Ball, A.; Bluche, T.; Caulier, A.; Leroy, D.; Doumouro, C.; Gisselbrecht, T.; Caltagirone, F.; Lavril, T. Snips voice platform: An embedded spoken language understanding system for private-by-design voice interfaces. *arXiv* **2018**, arXiv:1805.10190. [CrossRef].
41. Hemphill, C.T.; Godfrey, J.J.; Doddington, G.R. The ATIS spoken language systems pilot corpus. In Proceedings of the a Workshop Held at Hidden Valley, Stroudsburg, PA, USA, 24–27 June 1990; pp. 24–27.